AI-samples

This directory contains some sample codes for AI working on amnimo AI Edge Gateway.

In this device, you can use three DNN frameworks working with AI Accelerator Board:

- ONNX Runtime v1.2
- Tensorflow v1.15.2
- dv-sdk

For working with AI Accelerator Board, some part of sources of ONNX Runtime and Tensorflow are modified.

Sample Codes

ort_tiny_yolo_v3.py

This sample runs tiny Yolo V3 on ONNX Runtime.

To Setup:

1. download a model file:

```
~/ai-samples-master$ cd model
~/ai-samples-master/model$ wget
https://github.com/onnx/models/raw/master/vision/object_detection_segmentation/tin
y-yolov3/model/tiny-yolov3-11.onnx
```

Or you can put the file via SCP when you cannot connect GW with the internet directly.

2. put input images

Please make sure you put images (extension must be `.jpg`) to detect object to `object_detection_images/` subdirectory before running the script.

For example, you can download listed in `object_detection_images/images_index` by using `wget`.

To run:

```
~/ai-samples$ sudo python3 ./ort_tiny_yolo_v3.py
```

- inputs
  - Images in `object_detection_images/`
- outputs
  - Images added detected bounding boxes to the inputs to `outputs_ort_yolo/`

tf_keras_mobilenetv2.py

This sample runs MobileNet V2 on Tensorflow Keras.

To Setup:

Please make sure you put images (extension must be `.jpg`) to classify to `classification_images/` subdirectory before running the script.

For example, you can download listed in `classification_images/images_index` by using `wget`.

To run:

```
~/ai-samples$ sudo python3 ./tf_keras_mobilenetv2.py
```

- inputs
  - Images in `classification_images/`
- outputs
  - Strings of class label to standard output

cpp_yolov3_tiny

This sample runs tiny Yolo V3 using dv-sdk which is witten in C++.

To Setup:

Please download listed in `cpp_yolov3_tiny/images/images_index` by using `wget`, and rename files as indicated in the file, because in this sample, input file names are hard-coded in `cpp_yolov3_tiny/main.cpp`. So if you would like to use other file as input, please modify the source file.

Generate files by using Network Converter

This sample needs files named "cpp_yolov3_tiny*", which are generated from configuration file(.ini) and Keras standard model file(.h5) by Network Converter in a part of dv-sdk.

```
~/ai-samples/cpp_yolov3_tiny$ wget
https://github.com/DigitalMediaProfessionals/application/raw/master/model/yolov3-
tiny.h5
~/ai-samples/cpp_yolov3_tiny$ ls yolov3*
yolov3.ini  yolov3-tiny.h5
~/ai-samples/cpp_yolov3_tiny$ python3 /opt/amnimo-dv720/cnn_converter/convertor.py
yolov3.ini
~/ai-samples/cpp_yolov3_tiny$ ls cpp_yolov3_tiny*
cpp_yolov3_tiny_gen.cpp  cpp_yolov3_tiny_gen.h  cpp_yolov3_tiny_weights.bin
```

As input of Network Converter, you can use model files for Keras and Caffe.
For more details about Network Converter and dv-sdk samples, please see manual and application.

To build and run:

```
~/ai-samples$ cd cpp_yolov3_tiny
~/ai-samples/cpp_yolov3_tiny$ sudo apt update && sudo apt install build-essential
libopencv-highgui-dev
~/ai-samples/cpp_yolov3_tiny$ ls
cpp_yolov3_tiny_gen.cpp  cpp_yolov3_tiny_gen.h  cpp_yolov3_tiny_weights.bin
images  main.cpp  Makefile  output  YOLOv3_param.h  YOLOv3_post.cpp  YOLOv3_post.h
~/ai-samples/cpp_yolov3_tiny$ make
~/ai-samples/cpp_yolov3_tiny$ sudo ./cpp_yolov3_tiny
```

- inputs
  - Images in `cpp_yolov3_tiny/images/` subdirectory (file names are hard-coded in `cpp_yolov3_tiny/main.cpp`)
- outputs
  - Images added detected bounding boxes to the inputs to `cpp_yolov3_tiny/output/` subdirectory